

Estruturas de Dados - ESP412

Prof^a Ana Carolina Sokolonski

Bacharelado em Sistemas de Informação
Instituto Federal de Ciência e Tecnologia da Bahia
Campus de Feira de Santana

carolsoko@ifba.edu.br

December 5, 2024

Listas, Filas e Pilhas

1 Listas

- Listas Simplesmente Encadeadas

2 Referências

Listas

Listas Simplesmente Encadeadas

Buscar:

A função buscar consiste em percorrer a lista atrás da célula que contém o conteúdo específico que está sendo buscado. Assim, a função buscar deverá retornar um ponteiro para a célula da lista que contém o conteúdo de interesse. Esta função também pode ser construída de forma iterativa ou recursiva, vejamos as duas opções:

Listas Simplesmente Encadeadas

Buscar:

A função buscar consiste em percorrer a lista atrás da célula que contém o conteúdo específico que está sendo buscado. Assim, a função buscar deverá retornar um ponteiro para a célula da lista que contém o conteúdo de interesse. Esta função também pode ser construída de forma iterativa ou recursiva, vejamos as duas opções:

```
celula *buscar(int x, celula *listaEncadeada){
    celula *p;
    p = listaEncadeada;
    while (p != NULL && p->conteudo != x)
        p = p->prox;
    return p;
}
```

Listas Simplesmente Encadeadas

Buscar:

A função buscar consiste em percorrer a lista atrás da célula que contém o conteúdo específico que está sendo buscado. Assim, a função buscar deverá retornar um ponteiro para a célula da lista que contém o conteúdo de interesse. Esta função também pode ser construída de forma iterativa ou recursiva, vejamos as duas opções:

```
celula *busca_r(int x, celula *listaEncadeada){  
    if (listaEncadeada == NULL) return NULL;  
    if (listaEncadeada->conteudo == x) return listaEncadeada;  
    return busca_r (x, listaEncadeada->prox);  
}
```

Listas Simplesmente Encadeadas

Remover:

A função remover pode ser construída de uma forma simples, onde passa-se o ponteiro, por valor, do elemento anterior e deseja-se remover o próximo elemento:

Listas Simplesmente Encadeadas

Remover:

A função remover pode ser construída de uma forma simples, onde passa-se o ponteiro, por valor, do elemento anterior e deseja-se remover o próximo elemento:

```
void remover(celula *p){  
    celula *lixo;  
    lixo = p->prox;  
    p->prox = lixo->prox;  
    free (lixo);  
}
```


Listas Simplesmente Encadeadas

Buscar e Remover:

Ou pode-se informar o conteúdo a ser excluído. A função deverá buscar a célula correspondente e excluí-la:

Listas Simplesmente Encadeadas

```
celula *buscar_e_remover(int x, celula *listaEncadeada){
    celula *p, *q;
    p = listaEncadeada;
    if(p->conteudo==x){
        q=listaEncadeada;
        p=listaEncadeada->prox;
        free(q);
        return p;
    }else{
        q = listaEncadeada->prox;
        while (q != NULL && q->conteudo != x) {
            p = q;
            q = q->prox;
        }
        if (q != NULL) {
            p->prox = q->prox;
            free (q);
        }
        return listaEncadeada;
    }
}
```

Listas Simplesmente Encadeadas

```
int main(){
    int resp, resp2=1;
    printf("Lista Simplesmente Encadeada \n");
    celula *topo, *buscado;

    for(int j=0; j<10;j++){
        printf("Digite o próximo elemento da lista: ");
        scanf("%d", &resp);
        topo = inserirNoFinal(resp, topo);
        imprimir(topo);
    }
    printf("Fim da Insercao \n");

    while (resp2){
        printf("Digite o elemento que deseja encontrar na lista: ");
        scanf("%d", &resp);

        topo = buscar_e_remove(resp, topo);

        imprimir(topo);

        printf("Deseja repetir a busca? 1 - Sim, 0 - Não\n");
        scanf("%d", &resp2);
    }
}
```

Referências

Referências